Fermilab

Electrical Engineering Department --MS#222
Wilson Hall 14th Floor

To: Rick Ford, Bill Marsh

From: Craig Drennan

CC: Robert Goodwin

Date: 6/19/2002

Re: IRM Local Application for the MiniBoone Target BPMs

---

1. Introduction

We have developed new readout electronics for Beam Position Monitors to be used for the MiniBoone experiment. The readouts will only be used on the four BPM's nearest the target. The scheme is to demodulate the RF signals from each channel of a BPM and then digitize the demodulated signals using the 10 MSPS "Quicker Digitizer" available in the IRM, Internet Rack Monitor networked DAQ chassis.

This note describes the IRM local application program, written by Robert Goodwin, that manages the digitization of the four pairs of BPM signals and computes all four beam positions. The purpose of writing this note is mainly for my own future reference and an exercise to better understand how the local application works.

2. Execution of the IRM BPM Local Application, MBPM.c

Each time the Local Application (LA) is called two items are passed to the LA. The first is the variable `trig` which takes one of three values; `init, term, and cycle`. The action taken in response to the value of `trig` is explained in the following sections. The second item passed to the local application is a pointer to the list of parameters used by the LA. Table 2.1 lists these parameters. Some of the parameters are pointers to the first in a sequence of IRM channels. Table 2.2 and Table 2.3 expands the list of IRM channels that are referenced by the LA .

Table 2.1 MBPM Local Application Parameters.

Console variable name	**ParamList structure member name as defined in MBPM.c**	**Associated SM structure member name as defined in MBPM.c**	**Description of usage**
ENABLE B	short enableBit	_ _	This status bit enables or disables execution of the MBPM LA. The LA itself will set this bit low if an error occurs when trying to allocate the static memory structure.
#BPMS	short nBPM	short nBPMs	Specifies the number of BPMs that a position is computed for.
SIGNAL C	short sigChan	short sChanBase	*Not currently used*. This is the base channel that could point to the values taken directly from the digitizer memory for each digitizer channel.
POSITN C	short posChan	short pChanBase	This is the base IRM channel that points to the first of 32 IRM channels. The first four channels are the calculated BPM positions. Table 2.2 list the other values in this sequence of channels
CALIB C	short calChan	short cChanBase	This is the base IRM channel that points to the first of 16 IRM channels of calibration coefficients used in the BPM position calculations. Table 2.3 list these coefficients.
#BASE	short nBase	short baseNpts	Number of data points used in calculating the baseline value for each BPM channel.
#RISE	short riseOff	short rOff	Number of data points past the point where the rising edge is detected which will be the first data point used in the position calculation.
#PTS	short nPts	short nPts	Number of points summed for each digitizer channel and used in the position calculation.
EVENT# C	short eventNumChan	short eventNum (Holds event number not the channel number)	If the Event Number has a value between 0x01 and 0xFF it represents a Tev Clock event number. When the LA is called at the 15 Hz rate, the LA requires that the specified clock event has occurred. The LA returns immediately if it has not. If the Event Number is >= 0x0100 then the number is interpreted as a digital IO bit. If the bit is high the LA runs.
(spare)	short spar1	_ _	Spare parameter for future use.

Table 2.2 Output values referenced by the POSITN C base channel

Channel Number	ACNET Name	SM structure member name as defined in MBPM.c	Description
POSITN C	E:BPM1P	pos[0]	Calculated position for BPM 1
POSITN C + 1	E:BPM2P	pos[1]	Calculated position for BPM 2
POSITN C + 2	E:BPM3P	pos[2]	Calculated position for BPM 3
POSITN C + 3	E:BPM4P	pos[3]	Calculated position for BPM 4
POSITN C + 4	E:BPM1AV	avgdata[0]	BPM 1 average data computed as (SumA + SumB)/(2*Npts)
POSITN C + 5	E:BPM2AV	avgdata[1]	BPM 2 average data computed as
POSITN C + 6	E:BPM3AV	avgdata[2]	BPM 3 average data computed as
POSITN C + 7	E:BPM4AV	avgdata[3]	BPM 4 average data computed as
POSITN C + 8	E:BPM1AB	ped[0]	BPM 1 Channel A baseline value
POSITN C + 9	E:BPM1BB	ped[1]	BPM 1 Channel B baseline value
POSITN C + 10	E:BPM2AB	ped[2]	BPM 2 Channel A baseline value
POSITN C + 11	E:BPM2BB	ped[3]	BPM 2 Channel B baseline value
POSITN C + 12	E:BPM3AB	ped[4]	BPM 3 Channel A baseline value
POSITN C + 13	E:BPM3BB	ped[5]	BPM 3 Channel B baseline value
POSITN C + 14	E:BPM4AB	ped[6]	BPM 4 Channel A baseline value
POSITN C + 15	E:BPM4BB	ped[7]	BPM 4 Channel B baseline value
POSITN C + 16	E:BPM1AR	rise[0]	BPM 1 Channel A data index of the rising edge
POSITN C + 17	E:BPM1BR	rise[1]	BPM 1 Channel B data index of the rising edge
POSITN C + 18	E:BPM2AR	rise[2]	BPM 2 Channel A data index of the rising edge
POSITN C + 19	E:BPM2BR	rise[3]	BPM 2 Channel B data index of the rising edge
POSITN C + 20	E:BPM3AR	rise[4]	BPM 3 Channel A data index of the rising edge
POSITN C + 21	E:BPM3BR	rise[5]	BPM 3 Channel B data index of the rising edge
POSITN C + 22	E:BPM4AR	rise[6]	BPM 4 Channel A data index of the rising edge
POSITN C + 23	E:BPM4BR	rise[7]	BPM 4 Channel B data index of the rising edge
POSITN C + 24	E:BPM1SA	sums[0]	BPM 1 Channel A sum of N data points used for the position calculation.

POSITN C + 25	E:BPM1SB	sums[1]	BPM 1 Channel B sum of N data points used for the position calculation.
POSITN C + 26	E:BPM2SA	sums[2]	BPM 2 Channel A sum of N data points used for the position calculation.
POSITN C + 27	E:BPM2SB	sums[3]	BPM 2 Channel B sum of N data points used for the position calculation.
POSITN C + 28	E:BPM3SA	sums[4]	BPM 3 Channel A sum of N data points used for the position calculation.
POSITN C + 29	E:BPM3SB	sums[5]	BPM 3 Channel B sum of N data points used for the position calculation.
POSITN C + 30	E:BPM4SA	sums[6]	BPM 4 Channel A sum of N data points used for the position calculation.
POSITN C + 31	E:BPM4SB	sums[7]	BPM 4 Channel B sum of N data points used for the position calculation.

Table 2.3 BPM Position calibration coefficients referenced by the CALIB C base channel.

Channel Number	**ACNET Name**	**SM structure member name as defined in MBPM.c**	**Description**
CALIB C	E:BPM1M1	cf[0]	BPM 1 calibration coefficient m1.
CALIB C + 1	E:BPM1M2	cf[1]	BPM 1 calibration coefficient m2.
CALIB C + 2	E:BPM1M3	cf[2]	BPM 1 calibration coefficient m3.
CALIB C + 3	E:BPM1SX	cf[3]	BPM 1 sensitivity scale factor Sx.
CALIB C + 4	E:BPM2M1	cf[4]	BPM 2 calibration coefficient m1.
CALIB C + 5	E:BPM2M2	cf[5]	BPM 2 calibration coefficient m2.
CALIB C + 6	E:BPM2M3	cf[6]	BPM 2 calibration coefficient m3.
CALIB C + 7	E:BPM2SX	cf[7]	BPM 2 sensitivity scale factor Sx.
CALIB C + 8	E:BPM3M1	cf[8]	BPM 3 calibration coefficient m1.
CALIB C + 9	E:BPM3M2	cf[9]	BPM 3 calibration coefficient m2.
CALIB C + 10	E:BPM3M3	cf[10]	BPM 3 calibration coefficient m3.
CALIB C + 11	E:BPM3SX	cf[11]	BPM 3 sensitivity scale factor Sx.
CALIB C + 12	E:BPM4M1	cf[12]	BPM 4 calibration coefficient m1.
CALIB C + 13	E:BPM4M2	cf[13]	BPM 4 calibration coefficient m2.
CALIB C + 14	E:BPM4M3	cf[14]	BPM 4 calibration coefficient m3.
CALIB C + 15	E:BPM4SX	cf[15]	BPM 4 sensitivity scale factor Sx.

2.1 Initialization of the Local Application

When the LA is called with `trig` set to `init` a static memory structure (SM) is allocated and the LA parameters are recorded into this structure. After initialization the LA uses the parameter values stored in the SM. Periodically (after a fixed number of LA calls) the parameters in the SM are updated with those passed to the LA. A pointer to the SM is stored in the list of LA parameters so that after initialization the LA can be pointed to its SM each time it is called.

2.2 Termination of the Local Application

When the LA is called with `trig` set to `term` the static memory structure memory is de-allocated and the pointer to the old SM in the LA parameters is set to NULL.

2.3 Execution of the Local Application During a Normal Cycle

When the LA is called with `trig` set to `cycle` the LA performs its normal cycle tasks. These tasks are listed in Table 2.3.1.

Table 2.3.1 Normal cycle tasks.

Task	**Description**
1. Test for the T-Clk event or external bit trigger	The LA is called from a list of other local applications to be run every 67 ms (15 Hz). The LA only performs the following normal cycle tasks if the specified T-Clk event has occurred since the last time the tasks were performed. If the specified event is greater than or equal to 0x0100, an IRM external digital IO bit is tested to see if the normal cycle tasks should be executed. Section 3. describes further how the LA triggering event is specified.
2. Maintain software diagnostic counters and timers	Two diagnostic indicators are maintained and stored in the SM. SM structure member variable `cyclesSince` keeps a count of how many times the LA was called since the last time the normal cycle tasks were performed. The member variable `elapsedCalc` is set to the number of microseconds spent performing the cycle tasks.
3. Retrieve the digitizer waveform memory for all digitizer channels	SM structure member variable `short* wave[8]` is an array of eight pointers. The first points to the start of digitizer memory (Channel 0) at 0x70000000. The following pointers are separated by 0x4000 (16,384) addresses. When the digitized values are retrieve the numbers are store in the SM structure member array `short data[8][MaxSample]`.
4. Compute the baseline value of every BPM digitizer channel	The "zero input" baseline output of each BPM Synchronous Demodulator is measured as a average of #BASE points at the beginning of the digitizer data record (Note the first data point in the digitizer data record is not used). It is important to set the triggering of the digitizers so that the BPM pulse signal does not occur in the data record until after the necessary number of samples of the baseline are taken. Numerical filtering of the baseline values from beam pulse to beam pulse is also performed. The baseline values are stored in SM member array `short ped[8]` .
5. Determine the index offset into the data records where the rising edge of the demodulated pulse appears.	The precise timing between the digitizer trigger and the arrival of the beam at the BPM detector can vary a small amount. Therefore, the index offset into the data records of each digitizer channel where the rising edge of the demodulated pulse occurs needs to be determined. It is a fixed offset from this point in the data record where we begin summing values to use in the position calculation. Section 4. provides further discussion on how the rising edge is determined.
6. Compute the position of the beam for each of the four BPM's	#PTS number of digitizer data points starting at the data index found for the rising edge plus an index offset of #RISE points are summed for all BPM channels. For each BPM, position is computed as a function of the channel A sum and channel B sum, and three calibration coefficients and the BPM sensitivity scale factor. Section 5. provides further discussion on how the positions are computed.
7. The computed values are written to their associated IRM / ACNET channels	All of the computed values starting at the POSITN C base channel, as listed in Table 2.2 are updated.

3. T-Clk Events and Digitizer Triggers and Timing

The ACNET devices involved in controlling the timing and triggering of the LA and the digitizers are listed in Table 3.1. The LA is triggered by the Tev Clock Event specified by E:MBPMEV. The Digitizers that capture the pulse outputs of the Synchronous Demodulators are setup to be triggered by the Quicker Digitizer boards external trigger input. This input is connected to Channel 0 of the IRM's T-Clk Decoder / Timer module. The IRM Timer module generates the necessary trigger upon detecting the specified clock event plus the specified delay.

Table 3.1 ACNET devices used for timing and triggering.

ACNET Device Name	Description
E:MBPMEV	This is the event number, EVENT# C, as described in Table 2.1. The LA will not run unless this clock event has occurred or the specified bit number has been set.
E:BPDIGT	This is a delay in milliseconds from the occurrence of a timer event specified by E:BPDGEV and the channel 0 output of the IRM Timer Card. The delay setting has a resolution of one microsecond.
E: BPDGEV	This is the Tev Clock Event which triggers channel 0 of the IRM Timer Card.

4. Determination of the Demodulated Signals' Rising Edges

The precise timing between the digitizer trigger and the arrival of the beam at the BPM detector can vary a small amount. Therefore, the index offset into the data records of each digitizer channel where the rising edge of the demodulated pulse occurs needs to be determined. It is a fixed offset from this point in the data record where we begin summing values to use in the position calculation.

The index into the data where the rising edge is assumed to occur is the smallest index where the data value is greater than a threshold. The threshold is the midpoint or average between the baseline value and the maximum value in the data record for the particular digitizer channel of interest. The minimum threshold value is approximately 100 mV above the baseline. If there are no data points above the minimum threshold the position is not computed for the BPM.

5. Computation of the BPM Beam Position

Once the offset into the digitized data record for each channel is determined we add an additional offset, #RISE. The net offset into the data is where we wish to take the values that will be summed and used in the position calculation. This is illustrated in Figure 5.1.

The position calculation is shown in Equation 5.1. An explanation of how the calibration coefficients are determined is given in the following document.

"Establishing and maintaining the MiniBoone Target BPM Calibration", Craig Drennan, June 6, 2002, (filename: MiniBoone_Target_BPM_Calibration.doc)

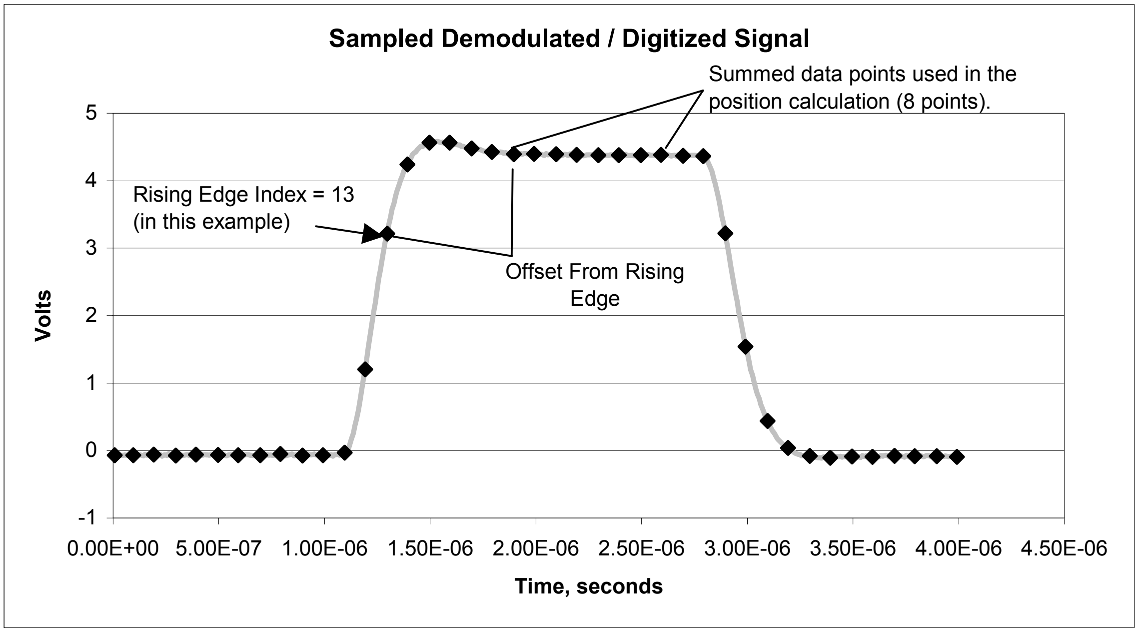

Figure 5.1 Illustration of which data points are used in the position calculation.

$$Pos\,(mm) = S_x \cdot \frac{m_1 \cdot \sum_{i=1}^{npts} A_i - \sum_{i=1}^{npts} B_i + (npts)\cdot(m_2 - m_3)}{m_1 \cdot \sum_{i=1}^{npts} A_i + \sum_{i=1}^{npts} B_i + (npts)\cdot(m_2 + m_3)} \qquad \text{Equation 5.1}$$

where,

S_x is the BPM Sensitivity value, 30.842.

$npts$ is the number of samples of the A and B channels (parameter #PTS).

$A_i,\ i = 1 \ldots npts$ Are the channel A data sample values.

$B_i,\ i = 1 \ldots npts$ Are the channel B data sample values

m_1, m_2, m_3 Are the calibration coefficients.